

Keaton Fox

Southern New Hampshire University

CS 470 – Full Stack Development II

18 December 2025

### CS 470 Final Reflection

This course helped me strengthen my understanding of cloud development by taking a full stack application from a local environment into AWS. I now have hands-on experience deploying and connecting core cloud services such as S3, API Gateway, Lambda, and DynamoDB. This work gave me a solid foundation in cloud-based application design and set me up to continue learning more advanced cloud development approaches in the future.

Through this course, I developed practical skills that make me a more marketable developer. These include containerizing applications with Docker, orchestrating services with Docker Compose, deploying a serverless frontend, building serverless APIs, and working with NoSQL databases in the cloud. I also gained experience applying IAM roles and policies to secure cloud resources. These skills are directly applicable to real-world cloud development and system design.

One of my main strengths as a software developer is my real-world perspective. I have built systems that are used globally, and I have also worked extensively as an end user of enterprise systems. This allows me to design solutions that are not only technically sound but also usable, scalable, and practical in day-to-day operations.

Based on my experience, I am prepared to continue working in a role that combines technical development with system ownership and collaboration. These include cloud-focused developer, systems lead, or solution-oriented technical roles. I also plan to continue building my own software to address multiple gaps I have found over the years.

This course reinforced how cloud services support future growth through scalability and flexibility. Microservices and serverless architectures make it easier to scale individual parts of an application without redesigning the entire system. Serverless services also reduce management effort by removing the need to maintain servers.

To handle scale and error handling, I would rely on managed AWS services that automatically adjust to demand. API Gateway and Lambda can scale based on usage, while logging and monitoring tools help identify and resolve errors quickly. This approach improves reliability without adding operational complexity.

Cost prediction in the cloud depends on understanding how services are used. Serverless pricing is based on actual usage rather than fixed capacity, which makes it easier to control costs for applications with changing demand. Compared to containers, serverless is generally more cost predictable for workloads that are not constant, while containers may make more sense for steady usage patterns.

When planning for expansion, tradeoffs must be considered. Serverless offers automatic scaling and lower maintenance, while containers provide more control but require more management. Elasticity and pay-for-service models make it possible to grow when needed and scale back when demand is lower. These principles reduce risk and allow the application to support growth in a controlled and cost-effective way.

Final Project Two Video: <https://www.youtube.com/watch?v=1HoFb8e-TtQ>